Leonardo Alves Pereira
João Paulo da Silva Cardoso

Desafio 4 – Aplicação da Cadeia Oculta de Markov

Candlesticks funcionam em sequências, certas sequências parecem determinar mudanças de preços. Se a classificação de candles do último desafio está correta teoricamente poderíamos aplicar Markov Hidden Model para determinar o impacto de certas sequências. Como? Qual seria o cálculo ? De um exemplo usando sua classificação anterior. O que o uso de MHM em sequência genética pode lhe ensinar.

Não serão aceitas respostas em código, apenas via explanação matemática, os números usados devem ser reais.

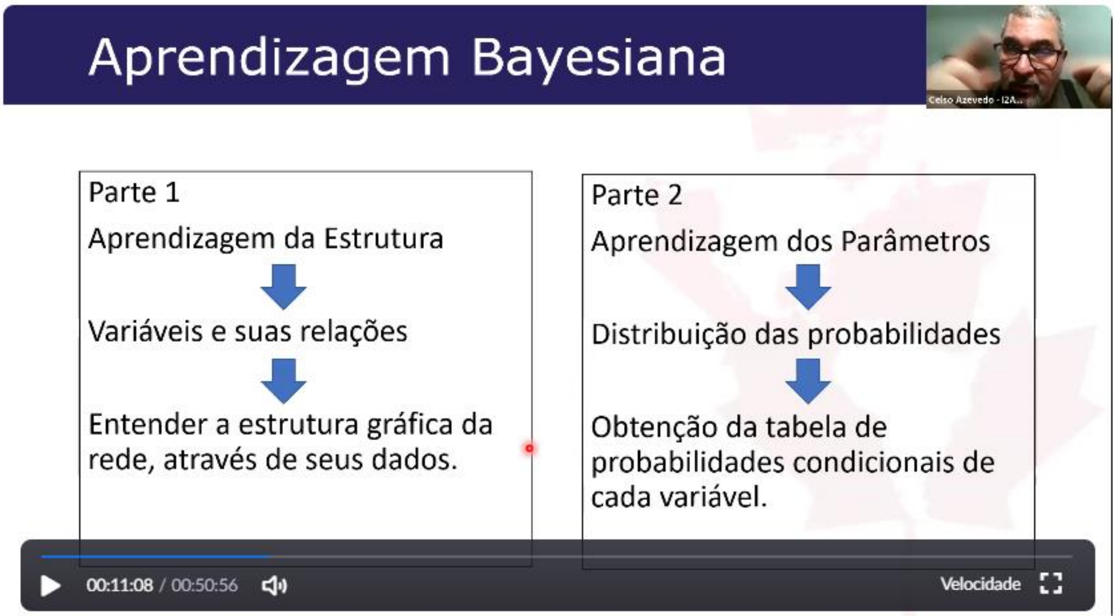
Entrega até próxima terça ao meio dia.

Sumário

1. Introdução

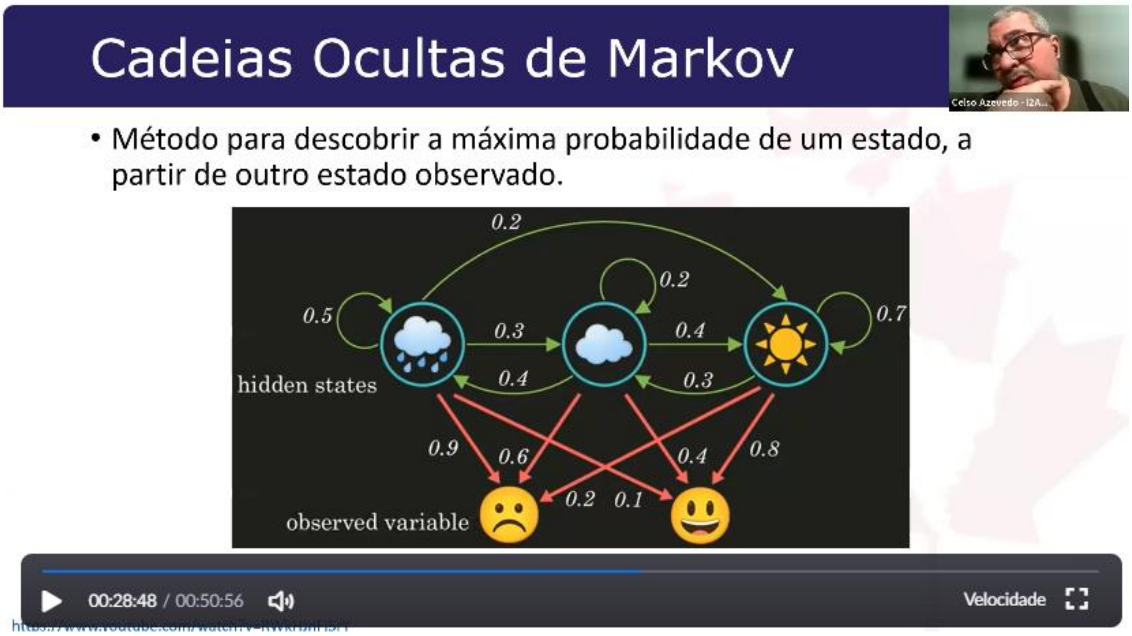
Vamos começar procurando associar o problema a alguns conceitos sobre cadeias bayesianas.

Da aula do Celso, vimos que:



Vamos então aplicar esses conceitos ao caso dos clusters de candlesticks

Um caso particular das redes bayesianas chamado de Cadeias Ocultas de Markov permitem calcular a probabilidade máxima de um estado a partir de um outro estado observado



---

2. Trabalhando com Cadeias Ocultas de Markov

Para trabalhar com as cadeias ocultas de Markov (Hidden Markov Models, ou HMMs) consideramos três matrizes principais para representar o modelo: A matriz dos "estados iniciais", a matriz das probabilidades de "transição" e as matrizes de "emissão" associadas ao estado verificável do processo:

2.1. Matriz de Transição de Estados (A)

Esta matriz descreve as probabilidades de transição entre os estados ocultos.

Cada elemento Aij da matriz representa a probabilidade de transição do estado i para o estado j:

$$\mathrm{Aij} = \mathrm{P}(\,\mathrm{St} = \mathrm{j} \mid \mathrm{St} - 1 = \mathrm{i}\,)$$

, onde:

S_t: Estado atual no tempo t.

As somas das probabilidades em cada linha devem ser iguais a 1 ($\sum_j A_{ij}=1$).

2.2. Matriz de Emissão (B)

Esta matriz relaciona os estados ocultos com as observações visíveis.

Cada elemento B_{jk} da matriz representa a probabilidade de observar o símbolo k dado o estado j:

$$\mathrm{Bjk} = \mathrm{P}(\,\mathrm{Ot} = \mathrm{k} \mid \mathrm{St} = \mathrm{j}\,)$$

, onde:
O_t: Observação no tempo t.
Cada linha de B também deve somar 1 ($\sum_k B_{jk}=1$).

2.3. Distribuição de Probabilidades Iniciais (π)

Esta matriz (ou vetor) define as probabilidades iniciais para cada estado oculto no tempo t=0.

Cada elemento $\pi_{i=}$ i representa a probabilidade de o sistema começar no estado i:

$$\pi i = P(S0 = i)$$

A soma de todas as probabilidades iniciais também deve ser 1 ($\sum_i \pi_i$=1).

2.4. Resumo das Matrizes:

Nome	Notação	Significado
Matriz de Transição	A	Probabilidades de transição entre estados ocultos.
Matriz de Emissão	B	Probabilidades de observação dado um estado oculto.
Probabilidades Iniciais	π	Distribuição inicial dos estados ocultos.

Essas matrizes juntas são usadas para calcular probabilidades em problemas típicos de HMM, como o cálculo da probabilidade de uma sequência de observações, a determinação da sequência mais provável de estados ocultos (algoritmo de Viterbi), e o ajuste dos parâmetros do modelo (algoritmo Baum-Welch).

2.5. Algoritmos usados com HMM

Principais algoritmos usados em problemas de **Cadeias Ocultas de Markov (Hidden Markov Models - HMMs)**, suas características, funções, aplicações e as bibliotecas Python que os implementam:

Algoritmo	**Características**	**Principais Funções**	**Aplicações**	**Bibliotecas Python**
Forward	Baseado em programação dinâmica. Calcula a probabilidade de uma sequência observada dado o modelo HMM.	Determinar P(O ∣ λ), onde O é a sequência observada e λ representa o HMM.	Reconhecimento de fala. Detecção de padrões de DNA. Modelagem probabilística de séries temporais.	hmmlearn, pomegranate, PyTorch, NumPy
Backward	Complementa o Forward. Calcula a probabilidade da sequência observada de trás para frente.	Eficiência em cálculos probabilísticos, especialmente para algoritmos de treinamento como o Baum-Welch.	Avaliação de probabilidades para observações futuras. Sistemas preditivos e de filtragem.	hmmlearn, pomegranate, NumPy

Algoritmo	**Características**	**Principais Funções**	**Aplicações**	**Bibliotecas Python**
Viterbi	Baseado em programação dinâmica. Encontra a sequência de estados ocultos mais provável.	Determinar P(S|O,λ), onde S é a sequência de estados ocultos mais provável.	Reconhecimento de fala. Modelagem de trajetórias. Análise de séries temporais categorizadas.	hmmlearn, pomegranate, PyTorch, TensorFlow
Baum-Welch	Algoritmo de Expectativa Maximização (EM). Treina os parâmetros do modelo HMM.	Estimar as matrizes A, B e π (transição, emissão e inicial) de um HMM dado um conjunto de dados.	Treinamento de modelos HMM em dados não supervisionados. Biologia computacional, análise de texto.	hmmlearn, pomegranate, scikit-learn
Decodificação Posterior	Usa probabilidades a posteriori para determinar o estado mais provável em cada ponto do tempo.	Determinar $\mathrm{argmax} P(St \mid O, \lambda)$, o estado oculto mais provável em t	Segmentação de texto. Análise de bioinformática. Modelagem de sistemas de controle.	hmmlearn, pomegranate, NumPy

2.5.1. Explicações adicionais:

- **Forward e Backward**:
  - Trabalham em conjunto para calcular probabilidades de forma eficiente.
  - São usados como base para o treinamento e análise de HMMs.
- **Viterbi**:
  - Um dos algoritmos mais famosos em HMMs, focado em encontrar a **sequência de estados ocultos mais provável**.
  - É amplamente usado em aplicações como tradução automática e reconhecimento de fala.
- **Baum-Welch**:
  - Essencial para **treinar HMMs**, especialmente quando os parâmetros do modelo (matrizes A, B, π) não são conhecidos.
  - Resolve problemas de aprendizado não supervisionado em cadeias ocultas de Markov.
- **Decodificação Posterior**:
  - Ao contrário do Viterbi, que foca em sequências inteiras, identifica o estado mais provável em cada instante de tempo.

2.5.2. Bibliotecas Python:

- **hmmlearn**: Simples e popular para modelagem HMM em Python. Suporta treinamento e decodificação.

- **pomegranate**: Biblioteca poderosa para modelos probabilísticos, incluindo HMMs. Mais flexível que hmmlearn.
- **NumPy**: Implementação manual de algoritmos, útil para aprendizado.
- **PyTorch/TensorFlow**: Usados para implementar HMMs otimizados com aprendizado profundo.
- **scikit-learn**: Contém algumas funções úteis que podem complementar HMMs.

Revisados esses conceitos, vamos agora interpretar mais profundamente qual é a nossa tarefa:

3. Tarefa do Desafio 4

---

4. Contextualizando o HMM no Desafio 4

Um HMM é adequado porque permite modelar uma sequência observável (*candlesticks*) associada a estados ocultos (*condições de mercado subjacentes*, como alta volatilidade, baixa liquidez ou reversão de tendência). Esses estados ocultos são determinados pelas transições entre clusters, o que pode ser analisado para prever mudanças de preço.

Os componentes do HMM neste caso específico são:

4.1. Estados ocultos (S):

Representam condições subjacentes do mercado. Suponha S={S1,S2,…,Sm}, onde cada estado é relacionado indiretamente aos clusters identificados pelo K-means.

4.2. Sequência observada (O):

É a sequência temporal de clusters atribuídos pelo K-means aos candlesticks. Exemplo: O={C2,C5,C3,C1,… }.

4.3. Matriz de transição (A)

Define as probabilidades de transição entre os estados ocultos, ou seja,

$$\mathrm{Aij} = \mathrm{P}(\,\mathrm{St} = \mathrm{j} \mid \mathrm{St} - 1 = \mathrm{i}\,)$$

4.4. Matriz de emissão (B):

Representa as probabilidades de observar um determinado cluster dado um estado oculto,

$$\mathrm{Bjk} = \mathrm{P}(\,\mathrm{Ot} = \mathrm{Ck} \mid \mathrm{St} = \mathrm{j}\,)$$

4.5. Distribuição inicial (π):

Define as probabilidades iniciais dos estados ocultos

$$\pi\mathrm{i} = \mathrm{P}(\mathrm{S0} = \mathrm{i})$$

---

5. Abordagem matemática

5.1. Determinar os estados ocultos (S)

Os estados ocultos S são derivados das transições de clusters (C) observados ao longo do tempo. Assumimos que cada estado tem uma relação estatística com os clusters, mas não sabemos exatamente qual.

5.2. Calcular as matrizes do modelo

Para estimar A, B, e π, precisamos:

Contagem de transições entre clusters para derivar A. Por exemplo, se do cluster C1 para C2 ocorrem 50 transições e, no total, partimos 200 vezes de C1, então:

$$\mathrm{A12} = \frac{50}{200} = 0.25$$

Frequência de associações entre estados ocultos e clusters observados para derivar B. Por exemplo, se S1 gera C3 em 60% dos casos, então:

$$\mathrm{B13} = 0.6$$

Proporção inicial de observações de cada cluster para estimar π. Por exemplo, se C1 é o cluster inicial em 30% das sequências, então:

$$\pi 1 = 0.3$$

5.3. Detectar padrões de sequência e impacto

Com o HMM parametrizado, utilizamos o **algoritmo de Viterbi** para identificar a sequência mais provável de estados ocultos associados a uma sequência observada de clusters. Isso permite inferir se certas sequências de *candlesticks* indicam uma mudança de preço ou tendência.

Exemplo de cálculo prático:

1. Suponha uma sequência observada O={C2,C5,C3}.
2. Dados A, B, e π, usamos o algoritmo de Viterbi para determinar a sequência de estados ocultos S mais provável.

5.4. Aplicações genéticas e lições aprendidas

A análise de sequências com HMM em genética se concentra em:

- **Identificar padrões regulares em dados sequenciais:**
  Similar aos padrões do DNA, podemos detectar padrões recorrentes em *candlesticks* que indicam reversões ou continuidade de preços.

- **Prever transições com base em estados ocultos:**
  Assim como mudanças de estados em genes indicam condições biológicas, mudanças em estados ocultos podem antecipar comportamentos de mercado.

5.5. Resultado esperado:

A aplicação de HMM deve revelar:

- **Sequências-chave de clusters** associadas a reversões ou consolidações.

- **Probabilidades preditivas** para mudanças de preço dadas as sequências observadas.

---

6. Aplicação ao nosso problema

No nosso Desafio 4, vamos adotar essa abordagem com HMM.

Temos um dataset com aproximadamente 6000 registros de candlesticks diários durante aproximadamente 20 anos. Já aplicamos a metodologia K-means e temos os resultados classificados em **5 clusters diferentes**.

A partir dessa base, vamos desenvolver passo a passo essa tarefa. Algumas proposições para realizar esse trabalho exigem recursos de computação (fora do escopo), mas serão relacionadas aqui, por exemplo: usar os clusters como observação, fazer com que o modelo aprenda as probabilidades de transição, definir os estados ocultos, testar as previsões para prever com segurança se o mercado vai mudar de estado. treinar o modelo usando algoritmos, obter a previsão probabilística dos cenários futuros para tomada de decisão, identificar os momentos favoráveis ou perigosos para investimento com base nos dados ocultos, escolher o número de estados ocultos por experiência prática ou usando métodos como validação cruzada ou BIC, analisar as emissões contínuas, treinar o HMM , validar o desempenho do modelo com dados de fora da amostra, e comparar com outras abordagens alternativas para confirmar a eficácia.

6.1. Etapa 1: Estrutura do Problema

1. Um **dataset** com 6000 registros de candlesticks diários.
2. Resultados de **K-means** classificados em 5 clusters.

A ideia é usar esses clusters como observações (input) para o modelo HMM e descobrir os estados ocultos (ex.: "alta", "baixa", "neutra") que descrevem as dinâmicas de mercado.

6.2. Etapa 2: Configuração Inicial

- **Bibliotecas necessárias**: Bibliotecas `hmmlearn` ou `pomegranate` em Python para implementar o HMM. Essas bibliotecas têm suporte para HMM com emissões discretas (clusters do K-means) ou contínuas (valores de candlesticks).

- **Estrutura do dataset**: Certifique-se de que ele está estruturado como:
  - Coluna de data (`Date`).
  - Coluna de clusters do K-means (`Cluster`).
  - Outros dados de interesse, como preços de abertura, fechamento, máxima e mínima (opcional).

6.3. Etapa 3: Usar os Clusters como Observações

Os clusters do K-means representam padrões no comportamento do mercado. No HMM:

- **Observações** são os clusters atribuídos pelo K-means.
- Exemplo: Se o K-means classificou os candlesticks em `5 clusters` (de 0 a 4), sua sequência de observações seria algo como:

observations = [0, 2, 1, 3, 4, 2, 1, 0, ...]

6.4. Etapa 4: Treinar o Modelo HMM

1. **Definir o número de estados ocultos**:
   - Comece com 3 estados ocultos (`"Alta"`, `"Baixa"`, `"Neutra"`).
   - Teste variações posteriormente (ex.: 2 ou 4 estados).
   - Use critérios como o **BIC (Bayesian Information Criterion)** para decidir o melhor número de estados.
2. **Baum-Welch**:
   - Este é o algoritmo usado para estimar as probabilidades de transição e emissão.
   - A biblioteca `hmmlearn` o executa automaticamente durante o treinamento.

6.5. Etapa 5: Inferir os Estados Ocultos

Depois de treinar o HMM, você pode inferir a sequência de estados ocultos correspondente às observações.

Cada valor em `hidden_states` corresponderá a um estado oculto identificado pelo HMM (ex.: 0 para "Alta", 1 para "Baixa", 2 para "Neutra").

6.6. Etapa 6: Avaliar Probabilidades de Transição

O modelo também calcula as probabilidades de transição entre estados:

Essa matriz mostra, por exemplo:

- A probabilidade de o mercado mudar de "Baixa" para "Alta".
- A probabilidade de permanecer no mesmo estado.

6.7. Etapa 7: Previsão Probabilística de Cenários Futuros

Use o HMM para prever probabilidades de transição para futuros estados:

Para probabilidades completas de todos os estados:

6.8. Etapa 8: Validar o Modelo

1. **Separar os dados**:
   - Divida os dados em `70%` para treino e `30%` para teste.
2. **Validar com dados fora da amostra**:
   - Use o modelo treinado nos dados de treino para prever os estados nos dados de teste.
   - Compare as previsões com os clusters ou movimentos reais do mercado.
3. **Métricas de validação**:
   - Log-likelihood: Mede a adequação do modelo aos dados.
   - BIC: Penaliza modelos mais complexos e ajuda a determinar o número ideal de estados.

6.9. Etapa 9: Identificar Momentos Favoráveis ou Perigosos

Analise os estados ocultos para associá-los a tendências de mercado:

- Identifique quais estados estão associados a altos retornos ou grandes quedas.
- Classifique os estados como "favoráveis" ou "perigosos" com base na análise dos dados históricos.

6.10. Etapa 10: Comparar com Abordagens Alternativas

- Compare o desempenho do HMM com modelos como LSTM ou Random Forest.
- Use backtesting para avaliar qual método maximiza retornos ou reduz perdas em um cenário real.

---

7. Resumo do Processo

1. Use os clusters do K-means como observações no HMM.
2. Defina o número de estados ocultos com base no problema.
3. Treine o modelo com Baum-Welch.
4. Inferira os estados ocultos e analise probabilidades de transição.
5. Valide com dados fora da amostra.
6. Associe estados a condições favoráveis/perigosas e teste em backtesting.

Estamos trabalhando em um problema de modelagem a partir de uma série temporal de resultados diários de investimentos de um tipo de ação. A partir dos aproximadamente seis mil registros diários de candlesticks, conseguimos agrupar esses dados em um número de clusters que supostamente contém candlesticks com características semelhantes. Como utilizamos uma análise não supervisionada, a partir do algoritmo K-means, podemos dizer que temos agrupamentos de candlesticks, mas não sabemos exatamente se esses agrupamentos foram bem classificados ou não, e sequer sabemos quais são as características desses agrupamentos quanto aos tipos conhecidos de candlesticks da literatura.

Pois bem, a nossa tarefa agora é: "Candlesticks funcionam em sequências, e certas sequências parecem determinar mudanças de preços. Se a classificação de candlesticks do último desafio está correta, teoricamente poderíamos aplicar Markov Hidden Model para determinar o impacto de certas sequências. Como? Qual seria o cálculo? Dê um exemplo usando sua classificação anterior. O que o uso de HMM em sequência genética pode lhe ensinar? Não serão aceitas respostas em código, apenas via explanação matemática, e os números usados devem ser reais.

Para abordar esse problema de maneira fundamentada, explicaremos como modelar a sequência de *candlesticks* utilizando um **Modelo Oculto de Markov (Hidden Markov Model - HMM)**, aproveitando os agrupamentos realizados anteriormente pelo algoritmo K-means. Vamos organizar os passos necessários e detalhar o raciocínio matemático por trás do uso do HMM para detectar padrões e impactos nas sequências.

---

8. Abordagem Alternativa

Embora não fosse escopo do Desafio 4, relacionamos aqui uma abordagem alternativa que nos pareceu relevante ao contexto do objetivo da aplicação do HMM.

8.1. Avaliação baseada em padrões temporais (análise de séries temporais)

- **Descrição**: Após a classificação por K-means, os clusters são utilizados para identificar padrões recorrentes ao longo do tempo. Podemos aplicar métodos como:
  - Moving Averages:
    Analise tendências dentro de cada cluster para identificar possíveis sinais de compra/venda.
  - Análise de autocorrelação:
    Verifique se há ciclos temporais ou dependências significativas em cada cluster.
  - Modelos ARIMA/Prophet:
    Modele previsões de valores futuros com base nas características de cada cluster.
- **Vantagem**: Permite associar os clusters a movimentos específicos no mercado e identificar padrões previsíveis.

8.2. Backtesting de estratégias por cluster

- Utilizamos os clusters para criar estratégias de investimento, simulando como essas estratégias teriam se saído no passado:
  - Definimos regras para entradas e saídas baseadas nos clusters (ex.: "Comprar quando entrar no Cluster 1, vender no Cluster 2").
  - Executamos backtests nos dados históricos.
- Métrica de Avaliação:
  - Sharpe Ratio: Retorno ajustado ao risco.
  - Drawdown Máximo: Risco de perda acumulada.
  - Percentual de acertos e payoff médio.
- Avaliamos diretamente a viabilidade prática dos clusters para decisões de investimento.

8.3. Avaliação de agrupamento com métricas de desempenho financeiro

- Comparamos os clusters em termos de métricas financeiras associadas, como:
  - **Retorno médio por cluster**: Qual cluster possui maior potencial de ganho?
  - **Volatilidade média por cluster**: Qual cluster é mais arriscado?
  - **Relação risco-retorno**: Identifique clusters com melhor perfil de investimento.
- Oferece insights sobre quais clusters são mais promissores para diferentes perfis de risco.

8.4. Redução dimensional e análise de transições

- Usamos métodos de redução dimensional (ex.: PCA, t-SNE) para visualizar os clusters em um espaço reduzido e entender como as transições entre clusters ocorrem ao longo do tempo.
- Criamos uma **matriz de transição de Markov** para calcular as probabilidades de mudança de um cluster para outro, permitindo prever os próximos estados.

- Fornece um mapeamento claro das dinâmicas de mercado e ajuda a antecipar movimentos.

8.5. Integração com machine learning para classificação e predição

- Utilizamos os clusters como novas features para treinar modelos preditivos mais avançados (ex.: LSTM, Random Forest):
  Alimentamos o modelo com clusters anteriores como contexto para prever futuros movimentos ou preços. Envolve aprendizado contínuo e maior adaptabilidade.

---

9. Como escolher a melhor opção?

A. **Se o objetivo é entender padrões temporais:**
Priorizamos as análises de séries temporais e a matriz de transição.

B. **Se o foco é criar e validar estratégias práticas:**
Escolhemos o backtesting de estratégias.

C. **Se queremos explorar insights gerais sobre os clusters:**
Analisamos as métricas financeiras ou reduzimos a dimensionalidade.

D. **Se queremos um modelo preditivo de longo prazo:**
Integramos os clusters em um modelo de machine learning.

O uso de **Cadeias Ocultas de Markov (HMM)** para avaliar uma sequência temporal de investimentos é uma abordagem muito poderosa, especialmente em cenários onde os estados subjacentes (como tendências de mercado) não são diretamente observáveis, mas podem ser inferidos a partir de observações, como os clusters identificados pelo K-means. Vamos explorar como isso pode se encaixar e os seus benefícios:

---

10. Observações sobre a aplicação das Cadeias Ocultas de Markov

O HMM é ideal para lidar com séries temporais que exibem dependências entre eventos. Aqui está como o método se relaciona ao nosso caso:

A. **Observações (emissões)**:

   Os clusters gerados pelo K-means (por exemplo, *Cluster 1, Cluster 2, etc.*) são usados como observações.

   Alternativamente, você pode usar os preços ou indicadores derivados dos candlesticks como emissões contínuas.

B. **Estados Ocultos**:

   Cada estado oculto representa uma condição subjacente do mercado, como "alta volatilidade", "tendência de alta", ou "tendência de baixa".

   O objetivo do HMM é inferir a sequência desses estados ocultos com base nas observações.

C. **Transições de Estado**:

   O modelo aprende as probabilidades de transição entre os estados (ex.: a probabilidade de passar de "alta volatilidade" para "tendência de alta").

   Essas probabilidades ajudam a prever a probabilidade de o mercado mudar para um estado específico no futuro.

D. **Treinamento do Modelo**:

   O modelo é treinado nos dados históricos usando algoritmos como Baum-Welch para estimar os parâmetros.

10.1. Benefícios do HMM no contexto de investimentos

- **Modelagem de dinâmicas de mercado**:

  O HMM captura as transições entre diferentes regimes de mercado, permitindo identificar padrões mais complexos do que simples clusters.

- **Previsão probabilística**:

  Em vez de uma previsão rígida, o HMM fornece a probabilidade de diferentes cenários futuros, permitindo uma tomada de decisão mais informada.

- **Identificação de regimes de mercado**:

  Pode ajudar a identificar "momentos favoráveis" ou "perigosos" para o investimento, com base nos estados ocultos.

- **Integração com K-means**:

Os clusters de K-means fornecem observações discretas que tornam o treinamento do HMM mais eficiente, especialmente se os dados originais forem contínuos.

10.2. Considerações práticas ao usar HMM

- **Escolha do número de estados ocultos**:
  - A escolha do número de estados (por exemplo, 3 para "alta", "baixa" e "neutra") é crucial e pode ser ajustada com validação cruzada ou critérios como o **BIC (Bayesian Information Criterion)**.
- **Análise de emissões contínuas**:
  - Se as emissões (candlesticks) forem valores contínuos, você pode usar distribuições gaussianas ou gaussianas mistas para modelá-las.
- **Treinamento com dados históricos**:
  - Treinar o HMM com dados suficientemente extensos para capturar diferentes regimes de mercado.
- **Validação e backtesting**:
  - Testar o desempenho do modelo com dados fora da amostra para garantir que ele generaliza bem.

10.3. Comparação com outras técnicas

- **HMM vs. Modelos Baseados em Regressão ou ML**:
  - HMM é mais interpretável quando se trata de estados ocultos, enquanto modelos como LSTM podem capturar padrões complexos, mas são caixas-pretas.
  - Se o seu objetivo for identificar regimes claros, HMM é a melhor escolha.
- **HMM vs. Simples Backtesting com K-means**:
  - O HMM agrega inteligência ao capturar as transições de mercado, enquanto o backtesting com clusters assume padrões estáticos.

10.4. Análise crítica (nossa)

O HMM é uma abordagem excelente neste contexto, especialmente porque ele:

1. Conecta os clusters do K-means (observações) a uma estrutura temporal mais rica.
2. Fornece insights interpretáveis sobre regimes de mercado.
3. Tem um histórico bem-sucedido em problemas financeiros.

Se o HMM for corretamente implementado, ele pode oferecer uma visão única dos ciclos de mercado e ajudar a otimizar as estratégias de investimento, desde que possamos garantir que seja feita uma validação rigorosa e comparações com outras abordagens alternativas para que possamos confirmar a eficácia da solução.
